УДК 004.853:025.4.036

Д.Г. Досин, В.М. Ковалевич
Фізико-механічний інститут ім. Г.В. Карпенка НАН України, м. Львів
Україна, 79060, м. Львів, вул. Наукова, 5

Архітектура інтелектуальної системи інформаційного пошуку в мережі Інтернет

D.G. Dosyn, V.M. Kovalevych
Karpenko Physical-Mechanical Institute
of the National Academy of Sciences of Ukraine, c. Lviv
Ukraine, 79060, c. Lviv, Naukova st., 5

Architecture of Intelligent System of Internet Information Search

Д.Г. Досин, В.М. Ковалевич
Физико-механический институт им. Г.В. Карпенко НАН Украины, г. Львов
Украина, 79060, г. Львов, ул. Научная, 5

Архитектура интеллектуальной системы информационного поиска в сети Интернет

У даній статті розглядається метод оцінки функції корисності природомовного тексту на основі його модельного представлення у формі повідомлення. Проведено аналіз стану досліджень та розробок у галузі інтелектуальних пошукових систем. Розроблено архітектуру інтелектуальної системи інформаційного пошуку в мережі Інтернет, яка функціонує як інтелектуальний агент.
Ключові слова: інтелектуальна система, інформаційний пошук, інтелектуальний агент.

In the given article, the method of estimating of utility function of natural language text based on its model representation in the form of massage is considered. State of research and investigations in intelligent information systems is provided. The architecture of intelligent system of Internet information search, which functions as an intelligent agent, is created.
Key Words: intelligent system, information search, intelligent agent.

В данной статье рассматривается метод оценки функции полезности естественноязыкового текста на основе его модельного представления в форме сообщения. Проведен анализ состояния исследований и разработок в области интеллектуальных поисковых систем. Разработана архитектура интеллектуальной системы информационного поиска в сети Интернет, которая функционирует как интеллектуальный агент.
Ключевые слова: интеллектуальная система, информационный поиск, интеллектуальный агент.

Вступ

Головною проблемою, що виникає в процесі інформаційного пошуку, є кількісна оцінка важливості для клієнта кожного зі знайдених документів – їх так званої релевантності. Метрика такої кількісної оцінки є локальною відносно масиву клієнтів, іншими словами, суб’єктивною: що становить інтерес для одного клієнта, може бути абсолютно неважливим для іншого. Мірою важливості може служити зміна ефективності стратегії клієнта. Якщо нове знання дозволяє клієнту по-новому оцінити свою стратегію і, можливо, змінити її, то знайдений новий документ важливий для цього клієнта. Щоб відрізнити цю характеристику від загальновживаної релевантності, назвемо її «потрібністю». Усі потрібні документи релевантні, але не всі релевантні документи клієнту потрібні.

Нам доводиться для цього припустити, що інформаційний пошук ведеться цілеспрямовано, тобто в інтересах того, хто має перед собою певну ціль і реалізує певну стратегію для її досягнення. Ціль, як і вся система понять клієнта, має міститися в онтології інтелектуальної системи, здатної представляти інтереси клієнта. Стратегія клієнта має бути представлена у базі знань інтелектуальної системи планом, побудованим в процесі машинного навчання системи. Онтологія та база знань органічно пов’язані між собою – план будується в системі координат, заданій онтологією. Таким чином, клієнт представлений в інтелектуальній системі інформаційного пошуку відповідним інтелектуальним агентом зі своєю онтологією та базою знань. Інформаційною моделлю клієнта є інтелектуальний агент після машинного навчання його бази знань та онтології як її складової.

Постановка задачі

Проблема автоматичного видобування знань передбачає оцінку важливості нових знань. Агентний підхід дозволяє нам оцінити «корисність» інформації. Визначимо знання як деякий узагальнений алгоритм (стратегія, метод) вирішення задачі, відображений як план виконання певної впорядкованої сукупності дій з переліку доступних чи допустимих. Кожна дія передбачає витрату ресурсів. Нагорода за вирішення задачі, як вартість бажаного стану агента, також вимірюється у розмірності ресурсів. Загальний виграш для агента – це різниця між витраченими та отриманими ресурсами. Таким чином визначається функція корисності відповідного алгоритму чи знання [1]. Якщо з’являється деяке нове знання, агент оцінює його функцію корисності і порівнює її до величин функції корисності для вже відомих алгоритмів. Її приріст (у розмірності ресурсів) пропонується вважати результуючою мірою новизни знань.

Необхідно розробити метод оцінки функції корисності природомовного тексту на основі його модельного представлення у формі повідомлення. Основні вимоги до методу – однозначність оцінки, незалежність від розміру текстового фрагмента, адаптивність та ефективність.

Раціональний інтелектуальний агент при виборі стратегії з переліку відомих з бази знань керується принципом максимальної очікуваної корисності з метою реалізувати своє призначення. Успіх на всіх стадіях реалізації обраної стратегії вимірюється через узагальнені ресурси як функція корисності. Якщо деякий текстовий документ описує новий алгоритм, можна отримати ненульову функцію корисності повідомлення, яка обчислюється агентом як різниця між відомим і новим виграшем.

Метою даної статті є розроблення архітектури інтелектуальної системи інформаційного пошуку в мережі Інтернет, яка функціонує як інтелектуальний агент, здатний вивчити інформаційні потреби клієнта, оцінити кількісно ефективність біжучої оптимальної стратегії цього клієнта та можливий приріст такої ефективності, якщо клієнт візьме до уваги зміст чергового повідомлення, потрібність якого встановлюється.

Стан проблеми

В галузі моделей і методів аналізу природомовних (ПМ) текстів на даний час отримані теоретично і практично значущі морфологічні моделі аналізу/синтезу лексем; розроблені моделі синтаксичного аналізу основних ПМ-конструкцій; запропоновано ряд методів реалізації основних моделей аналізу ПМ-конструкцій, які можуть використовуватися на практиці; виділені основні прийоми евристичної реалізації окремих моделей інтерпретації ПМ-висловлювань; опрацьовані окремі моделі концептуального синтезу ПМ-текстів; запропоновані і практично перевірені моделі і методи лінгвістич-

ного синтезу. В галузі моделей розуміння розроблені багаторівневі моделі, що враховують не тільки лінгвістичні, а й когнітивні складові цього процесу. В галузі реалізації розроблені прототипи інтелектуальних ПМ-систем; є промислові реалізації ПМ-систем різного класу, які, проте, в більшості випадків лише «імітують» повномасштабне розуміння природної мови [2].

Таблиця 1 – Передові групи розробників ПМ-засобів

1.	Thomas Watson Research Center
2.	Palo Alto Research Center
3.	Teragram
4.	Група опрацювання ПМ Стенфордського університету
5.	Американсько-ізраїльська компанія ClearForest
6.	German Research Center for Artificial Intelligence DFKI (Ontoprise GmbH)
7.	Natural Language Processing Research Group within the Department of Computer Science at the University of Sheffield

Таблиця 2 – Розробники практично реалізованих ПМ-засобів

1.	BASIS Tech	http://www.basistech.com
2.	ClearForest Corporation	http://www.clearforest.com
3.	CognIT	http://www.cognit.no
4.	Compris Intelligence GmbH	http://www.kompass.com
5.	Convera (formerly Excalibur)	http://www.convera.com
6.	Delphes	http://www.delphes.com
7.	Megaputer Intelligence Inc.	http://www.megaputer.com
8.	Insightful Corporation & InFact	http://www.insightful.com
9.	Inxight Software Inc.	http://www.inxight.com/
10.	MITRE	http://www.mitre.org
11.	Ontotext	http://www.ontotext.com
12.	SRA International Inc.	http://www.sra.com
13.	TEMIS	http://www.temis.com

Розроблено цілий ряд ефективних інструментів опрацювання ПМ текстових документів, проте на даному етапі існуючі методи комп'ютерної лінгвістики, штучного інтелекту, інформаційних технологій в цілому досі не мають ефективних моделей текстів на природних мовах, придатних для автоматичної чисельної оцінки їх потрібності (корисності) потенційному споживачу.

Оцінка ефективності

В основі бази знань інтелектуального агента лежить план, який є ієрархічним і на кожному рівні ієрархії містить альтернативний набір типових (звичних, перевірених досвідом) стратегій – маршрутів у просторі станів агента. Кожний із таких маршрутів на всіх етапах переходів зі стану в стан містить кількісні оцінки цінності станів та очікуваних затрат на переходи. Цінність станів взаємопов'язана через цінність інших станів та очікувані затрати на перехід до них. Завдяки такому представленню план дозволяє оцінювати на кожному рівні деталізації поведінки і кожному рівні прийняття рішення ефективність стратегії, як приріст цінності стану після переходу зі стану в стан, приведений до затрат на такий перехід (апостеріорна ефективність):

$$E_{ff} = \left(U(S_{i+1}) - U(S_i)\right) / R(A_{ik}), \qquad (1)$$

де $U(S_i), U(S_{i+1})$ – загальна (з врахуванням можливих подальших винагород) цінність (корисність, *utility*) станів біжучого та наступного після виконання *k*-ї дії з *i*-го стану A_{ik}; $R(A_{ik})$ – затрати на реалізацію переходу зі стану S_i в стан S_{i+1}. До реалізації такого переходу інтелектуальний агент бере до уваги оцінку імовірності $P(S_i, A_{ik}, S_j)$ досягнення стану S_j зі стану S_i при виконанні дії A_{ik}:

$$U(S_i) = R(S_i) + \gamma \cdot \max_{A_{ik}} \sum_k P(S_i, A_{ik}, S_j) U(S_j), \tag{2}$$

де $R(S_i)$ – так звана «короткотермінова», безпосередня, пряма винагорода при досягненні стану S_i; γ – коефіцієнт знецінювання, який набуває значення у діапазоні $\overline{0 \ldots 1}$. Цю оцінку тому можна вважати апріорною ефективністю стратегії агента.

Рівняння (2), яке називається рівнянням Белмана, покладено в основу алгоритму пошуку оптимальної стратегії побудови плану діяльності інтелектуального агента π^*, що застосовується для розв’язування таких задач, сформульованих, як MDP (Markov Decision Process) чи POMDP (Partly Observable Markov Decision Process), методами ітерацій за значеннями, ітерацій за стратегіями і т.п. [1].

Для побудови плану агент отримує інформацію про:

1. Множину станів S_i;
2. Імовірності тих чи інших наслідків дій $P(S_i, A_{ij}, S_j)$;
3. Множину джерел інформації з відповідними їм достовірностями $P(I_m)$;
4. Кінцеву мету діяльності, як множину ознак бажаного стану;
5. Вартість затрат на виконання кожної з множини можливих дій $R(A_{ik})$ з врахуванням залежності цієї вартості від стану, в якому вчиняється дія;
6. Множину ресурсів $R_l(S_i)$, доступну у кожному з можливих станів для їх застосування на вчинення якоїсь дії.

Таким чином, модель світу для інтелектуального пошукового агента являє собою шестиарний кортеж: $< S_i, P(S_i, A_{ij}, S_j), P(I_m)$, S_t, $R(A_{ik}), R_l(S_i) >$.

Особливість даного підходу полягає у тому, що виграш від переходу у деякий стан визначається приростом доступних агенту ресурсів, які в свою чергу можуть бути ним використані для вчинення дій для подальших переходів і по суті є вектором ресурсів. Таку модель назвемо **ресурсною моделлю**.

На відміну від традиційної MDP чи POMDP моделі, в якій задано (переважно скалярний) виграш від переходу у певний стан і приймається адитивна оцінка інтегрального виграшу, а затрати на виконання переходів закладені у виграш шляхом віднімання від виграшу цих затрат, у даному підході затрати виражені явним чином як затратний ресурс агента, без якого агент не може виконати дію та здійснити перехід в інший стан, а тому цей ресурс служить додатковим обмежуючим фактором при виборі наступної дії.

$$U(S_i) = R(S_i) \cdot R(A_{i-1,k}) + \gamma \cdot \max_{A_{ik}} \sum_k P(S_i, A_{ik}, S_j) U(S_j). \tag{3}$$

Крім того, на відміну від традиційної моделі, дана модель допускає на кожному кроці (такті) функціонування одночасне виконання деякої підмножини дій A_{ik*} з множини *k* можливих у даному стані $A_{ik*} \subseteq A_{ik}$, $k* \le k$. Враховуючи це, при виборі оптимальної стратегії інтелектуальний агент на кожному кроці свого функціонування

вирішує задачу вибору підмножини оптимальних дій з множини можливих з врахуванням біжучого обмеження на ресурси:

$$U(S_i) = \sum_{k^*} R(S_i)\cdot R(A_{i-1,k}) + \gamma \cdot \max_{A_{ik}} \sum_{k} P(S_i, A_{ik}, S_j)U(S_j), \quad \sum_{k^*} R(A_{i-1,k}) \le R_{S_i \max}. \qquad (4)$$

Отже, в такій постановці задача вибору оптимальної стратегії зводиться до задачі динамічного програмування, а саме до задачі про ранець. Для вирішення задачі вибору оптимальної стратегії на першому етапі агент досліджує оптимальну стратегію вищого рівня, розглядаючи множину можливих одночасних дій як одну макродію, мета якої – глобальна ціль функціонування агента. Після того як рішення щодо вчинення певної макродії прийняте, агент розкладає її на складові і намагається оптимізувати кожну з них, тобто вибирає стратегії нижчого рівня. Коли дії нижчого рівня обрано, агент вибирає методом рішення задачі про ранець – які з необхідних дій він зможе вчинити на даному кроці, враховуючи наявні обмеження у ресурсах. Якщо такі обмеження не дозволяють виконати достатній для реалізації макродії об’єм і перелік мікродій, сама макродія переглядається та обирається більш реалістична глобальна мета. Ця рекурсивна процедура може містити довільну кількість рівнів і бути реалізованою шляхом HTN-планування.

Вибір ключових елементів архітектури

Перш за все, необхідно визначитися з предметною областю (ПО), в якій інтелектуальна система (ІС) має бути компетентна. Крім того, необхідно визначитися з життєзабезпечуючим середовищем (ЖЗС), в якому має функціонувати дана ІС. Структура та функції ІС однозначно визначаються її ЖЗС.

Визначимо конкретне ЖЗС: це операційна система (ОС) Linux, Інтернет з його інформаційними службами нижнього рівня та клієнтами – потенційними замовниками послуг, інші інтелектуальні агенти (ІА) – конкуруючі служби інтелектуального інформаційного пошуку.

Визначимо ПО: виходячи з практичних міркувань, конкретних умов виконання досліджень і, як результат, з потреб замовника інформаційних послуг, це – *матеріалознавство*. Конкретніше – методи та засоби продовження ресурсу металоконструкцій об’єктів тривалої експлуатації сфери промисловості та транспорту.

ЖЗС та ПО дають нам вихідні дані для побудови ІА – інтелектуальної інформаційно-пошукової системи (ІІПС).

Вся БЗ – це HTN-план з відповідними даній ПО оцінками очікуваної корисності кожного етапу, кожної операції такого плану, цінності кожного стану, в який має перейти агент. Розробка адекватного методу оцінювання станів та операцій становить важливий (якщо не ключовий) етап розробки ІІПС.

Перш за все, ЖЗС ІІПС дає оцінку кінцевого стану ІІПС, адже за все платить кінцевий користувач. (Приклад такого розрахунку дає Google Analytics.) Інші розрахунки виконуються шляхом регресивного планування [1, с. 524] від цільового стану до біжучого. Якщо навіть кінцевий стан невизначений, ІА проектує кінцеву ціль на близьку:

$$U(C^*) = U(C)\cdot P(C \mid C^*), \qquad (5)$$

де C – далека ціль, C^* – близька ціль, $P(C|C^*)$ – умовна імовірність досягнення кінцевої цілі при умові досягнення близької цілі. Таким чином будується мережа довіри (Байєса) між цілями на різних етапах плану ІА.

За умови наявності оцінок цінності (корисності) станів плану може бути розрахована очікувана корисність дій ІА на кожному з етапів плану. Враховуючи, що план ієрархічний, розрахунок виконується, починаючи з найвищого рівня.

Аналіз стану досліджень та розробок в галузі інтелектуальних інформаційних систем та Інтернет-послуг [2] дає підстави вважати, що виправданими є наступні програмно-технічні рішення:

– реалізація ІІПС як служби портальної Інтернет-системи;

– застосування **OWL** як мови програмування онтології ІІПС;

– застосування **HTN** та OWL-S як структури та мови ПЗ автоматичного планування бази знань (БЗ) ІІПС;

– **Java API для Protege-OWL** – як програмного середовища та бібліотеки класів опрацювання, зокрема, машинного навчання (навчання з підкріпленням) OWL-онтології та БЗ ІІПС;

– **Link Grammar Parcer** – як засобу граматично-семантичного аналізу англомовних текстових документів в електронному форматі;

– **Apache-PHP-MySQL** – як програмних засобів для побудови інтерфейсу з користувачем за архітектурою веб-порталу;

– **Wget** – як веб-служби для автоматизованого доступу до пошукових серверів зі сформованим з ключових слів запитом;

– **SWRL** – апарат логічного виводу нових знань дедуктивним та індуктивним методами;

– **WordNet** – як базовий тлумачний словник англійської мови.

Онтологія ІІПС (на мові OWL) містить понятійний апарат верхнього рівня та предметної області матеріалознавства, описаної раніше. Онтологія верхнього рівня забезпечує:

– логічний вивід нових знань, доповнення отриманих повідомлень контекстом;

– верифікацію істинності отриманих тверджень;

– оцінку вірогідності джерел повідомлень;

– забезпечення логічної цілісності БЗ.

Основою онтології верхнього рівня служать предикати логіки 1-го порядку. Світ не вважається замкнутим, тобто твердження, що не випливають з БЗ, не вважаються хибними. Їх значення невизначене. Онтологія верхнього рівня (ОВР) містить апарат HTN, зокрема, поняття алгоритму (сценарію) поведінки, ресурсів та затрат, цінності стану, вектора ознак стану, очікуваної корисності алгоритму. ОВР містить також розділ граматичних форм семантичних залежностей між поняттями для технічної англійської мови, засоби машинного навчання на прикладах (тт. з вчителем), а також на основі логічного виводу методом індукції (узагальнення часткових випадків появи типових граматичних конструкцій.

Онтологія ПО формується з використанням засобів ОВР, тобто проблема ресурсу об’єктів тривалої експлуатації описується через алгоритми оцінки та продовження ресурсу, вартість об’єкта, затрати на підтримку його функціонування та продовження ресурсу експлуатації, очікувану корисність від застосування відповідних алгоритмів. Початкові структури онтології ПО формуються вручну. На наступному етапі для розбудови онтології застосовуються методи машинного навчання та відповідні навчальні тексти зі спрощеною граматикою та семантикою.

Машинне навчання реалізується засобами Java Protege-OWL API. Ці засоби містять бібліотеки класів, в яких реалізовано методи роботи з OWL-структурами: їх читання, доповнення. Таким чином, засоби машинного навчання (ЗМН) функціонують

у взаємодії з OWL-онтологією, беручи з неї шаблони граматично-семантичних структур для розпізнавання тверджень (предикатів логіки 1-го порядку) у досліджуваних і/або навчальних текстах та додаючи до неї нові елементи в результаті такого розпізнавання. Для цього застосовується Link Grammar Parcer (LGP), який розбиває стверджувальне речення, написане граматично правильною англійською мовою, на семантично пов'язані між собою пари слів (понять). LGP містить у своєму складі таблицю відповідності між граматичними конструкціями англійської мови та типами синтаксично-семантичних зв'язків між словами (поняттями). API LGP дозволяє пов'язати цю таблицю з OWL-онтологією, завдяки чому таблиця може динамічно адаптуватися в процесі навчання до заданої ПО.

ЗМН на базі Java-бібліотек Protege-OWL містить узагальнений опис семантичного зв'язку, який служить шаблоном для генерування в процесі навчання нових типів семантичних зв'язків та формування для їх ідентифікації в тексті відповідних векторів ознак цих зв'язків. При цьому до ОВР додаються відповідні класи зв'язків та їх властивості. Екземпляри цих класів служать для опису існуючих та нових класів онтології шляхом їх використання як предикатів логіки 1-го порядку.

Моделювання процесу планування

В роботі використовується середовище планування та інженерії знань загального призначення без прив'язки до конкретної предметної області, яке забезпечує побудову систем планування, базованих на OWL-онтологіях, та інтегрування процесу планування з логічним виводом, базованим на описовій логіці (description logic – DL) [3].

У моделі процесу планування стан світу описується множиною OWL-фактів. Вони представлені як RDF-граф [4]. Дії описуються через трансформації RDF-графів [5]. Цей метод планування дозволяє також генерувати оптимальні плани відповідно до адитивної метрики якості, коли кожна дія асоціюється з вектором якості і затратами на виконання.

Такий планувальник використовує логічний вивід на базі описової (DL) логіки за двохетапним алгоритмом: на першому етапі в режимі «off-line» планувальник транслює DL-описи дій на мову SPPL, в процесі чого виконує необхідний логічний вивід і додає необхідні факти до множини результатів планованих дій. На другому етапі використовується сам планувальник SPPL [6].

В описовій логіці використовуються два типи фактів: термінологічні – TBox та факти предметної області – ABox. Множина RDF-термів (RDF_T) включає в себе множину URI (U) та множину RDF-літералів (RDF_L). RDF-триплет є елементом множини $RDF_T \times U \times RDF_T$. RDF-граф є множиною RDF-триплетів. Вузли графа є суб'єктами і об'єктами, а ребра позначаються як властивості.

Застосована в роботі система підтримує підмножину OWL-DL, яка має назву Description Logic Programs (DLP) [7]. DLP є перетином DL та Horn Logic Programs. Модель світу подається RDF-графом, який містить триплети, що відображають OWL ABox факти. Модель дії базується на ідеї, що дія має спричиняти зміну стану світу. Передумови та результати моделюються як патерни RDF-графа. Дія виражається через трансформацію RDF-графа. Унарний предикат (Пристрій ?A1) на RDF-графі виражається через вузол екземпляра ?A1, зв'язаний ребром «типу» з вузлом класу «Пристрій». Бінарний предикат відображається на RDF-графі через два вузли екземплярів, з'єднаних ребром властивості.

Формально опишемо модель дії:

Змінна є елементом множини V, де V – необмежена і несумісна з RDF_T. Триплетний патерн є елементом множини ($RDF_T \cup V$) × U × ($RDF_T \cup V$). Патерн графа – це множина триплетних патернів. Дія має форму A(P,E,C,Q), де:

P – патерн RDF-графа, що виражає передумови дії; E – патерн RDF-графа, що виражає результати дії. Множина змінних в P є підмножиною множини змінних у E. Це гарантує відсутність невизначених змінних у результуючому описі. C – це вектор вартості дії; Q – це вектор якості (ефективності) дії.

Дія може бути застосована, якщо у даному стані виконуються всі передумови. Розглянемо дію *A*(P,E,C,Q). Нехай біжучим є стан G. Визначимо, що *P* може бути застосована до G, базуючись на онтології O, тоді і тільки тоді, якщо існує функція заміни змінних ($\theta: V \rightarrow RDF_T$), визначена для всіх змінних у P так, що: $G \cup O \models \theta(P)$.

Результат застосування дії описується як результат трансформації RDF-графів. Нехай **L** та **R** – передумови та результати дії відповідно, **L** задовольняє біжучий стан, описаний RDF-графом **X**, θ – функція заміни змінних в **L**. Нехай також наступний після застосування дії стан описується RDF-графом **Y**. Кожний **Y** визначається шляхом застосування гомоморфізму графів *f*, описаного наступним чином:

$$f : \theta(\mathbf{L}) \cup \theta(\mathbf{R}) \rightarrow \mathbf{X} \cup \mathbf{Y},$$

де *f* задовольняє наступні властивості:

1. $f(\theta(\mathbf{L})) \subseteq \mathbf{X}$
2. $f(\theta(\mathbf{R})) \subseteq \mathbf{Y}$
3. $f(\theta(\mathbf{L})\backslash\theta(\mathbf{R})) = \mathbf{X}\backslash\mathbf{Y}$ та $f(\theta(\mathbf{R})\backslash\theta(\mathbf{L})) = \mathbf{Y}\backslash\mathbf{X}$

Це значить, що саме та частина графа **X** вилучена, котра відповідає елементам θ (**L**), але немає в θ (**R**), а також саме та частина графа **Y** створюється, яка відповідає новим елементам в θ (**R**).

Застосовуючи властивості 2 і 3, можна визначити наступний стан **Y** як результат застосування дії до **X**. Операція виконується за два кроки. На першому кроці усуваються всі вузли і ребра з **X**, які відповідають $\theta(\mathbf{L})\backslash\theta(\mathbf{R})$, щоб отримати граф **D**, для якого $\mathbf{D}=\mathbf{X}\backslash(\theta(\mathbf{L})\backslash\theta(\mathbf{R}))$. Впевнюємося, що **D** – коректний граф, тобто не має висячих вузлів після видалення вихідних чи вхідних ребер. На другому кроці ми склеюємо **D** з **R\L**, щоб отримати **Y**. Інші деталі стосовно трансформації графів можна знайти в [8]. Мета планування подається як патерн RDF-графа. План досягає мети, якщо він приводить до кінцевого стану, який задовольняє патерн RDF-графа цієї мети.

Тестові моделі

Розглянемо приклад прийняття рішень ІДС стосовно модернізації трубопроводу (рис. 1). Початковий стан: *Необроблена*. Кінцевий стан (стан мети): *Оброблена*.

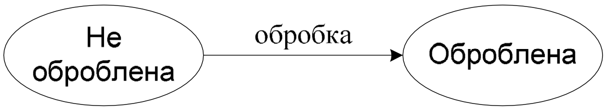


Рисунок 1 – Загальна задача модернізації трубопроводу

Задача ділиться на 3 підзадачі (*підготовка, покриття, захист*), перша з яких ділиться ще на 4 підзадачі (*розкриття поверхні труби, зняття захисного покриття,*

знежирення, *ґрунтування*), як показано на рис. 2. Для розв’язування кожної підзадачі використовуються альтернативні рішення. Так, для підзадачі *зняття захисного покриття* можна використати одну із трьох альтернатив: механічне, хімічне, термічне. Вся ця інформація зберігається у відповідній онтології (онтологія ПО модернізації нафто- та газопроводів у даний час перебуває у процесі розробки в лабораторії системного аналізу науково-технічної інформації Фізико-механічного інституту ім. Г.В. Карпенка НАН України) [9].

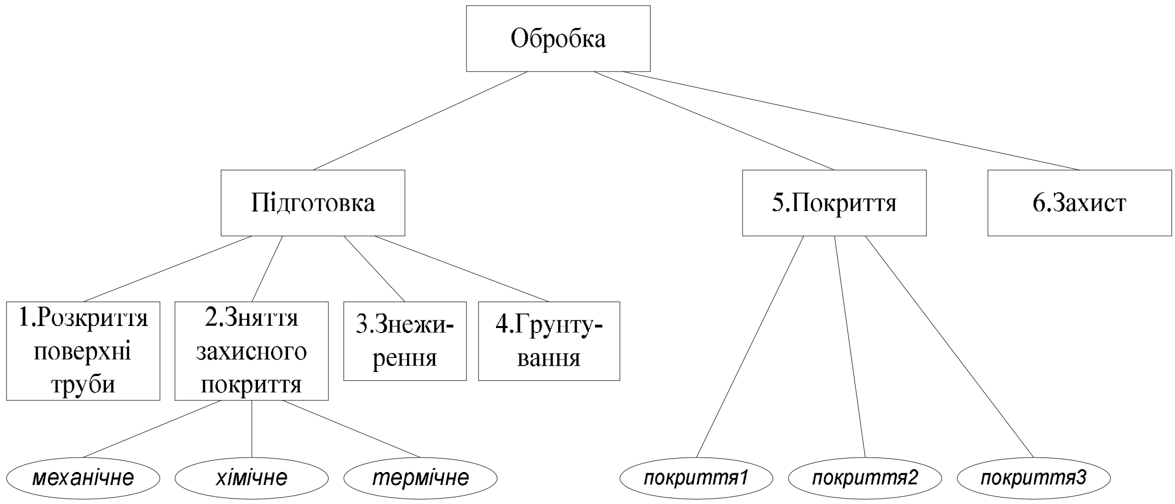


Рисунок 2 – Декомпозиція задачі «Обробка»

Отже, загалом необхідно послідовно розв’язати 6 підзадач $P_1, P_2, ..., P_6$. Для кожної задачі необхідно вибрати метод розв’язку (альтернативу). Якщо *G* – наявний ресурс, r_e – бажаний термін експлуатації трубопроводу, то раціональність прийняття рішень полягатиме в:

$$\begin{cases} U = \sum_{i=0}^{N-1} U\left(a_{ij}^{k}\right) \to \max, \\ r \ge r_e, \\ \sum_{i=0}^{N-1} g_{ij}^{k} \le G. \end{cases} \quad (3)$$

Нехай ресурс складає 6 умовних одиниць: *G*=6. Приклади можливих витрат g та виграшів U залежно від номеру процесу та альтернативи наведені у табл. 3.

Таблиця 3 – Таблиця витрат та доходів

№	Процес 1		Процес 2		Процес 3		Процес 4		Процес 5		Процес 6	
альтер.	Витр.	Дох.	Витр.	Дох.	Витр.	Дох.	Витр.	Дох.	Витр.	Дох.	Вит.	Дох.
a_1	0	5	0	8	0	3	0	4	0	2	0	3
a_2	1	7	1	9	1	4	1	7	1	3	1	7
a_3	2	8	2	12	-	-	-	-	2	6	-	-

Використовуючи метод функціональних рівнянь [4], призначений для розв’язування задач динамічного програмування, отримаємо оптимальний шлях, який наведений на рис. 4.

Оптимальний план розподілення ресурсів між процесами обробки трубопроводу наведений у табл. 4. Дохід від функціонування трубопроводу складе 40 одиниць.

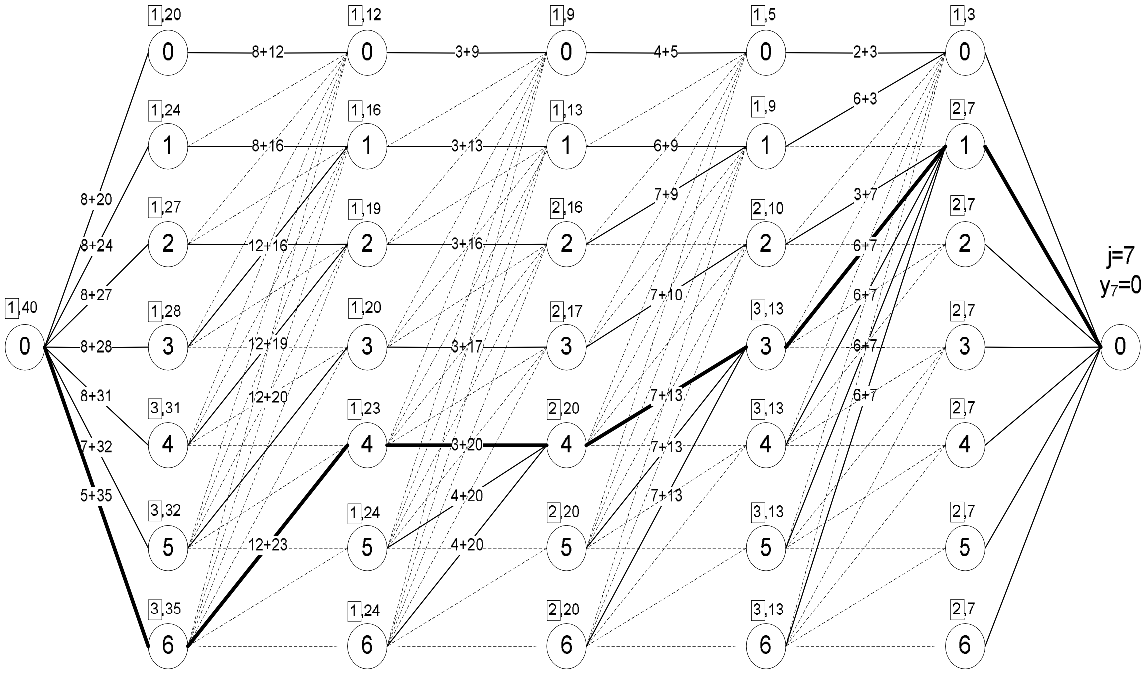


Рисунок 3 – Процес розв’язування задачі динамічного програмування

Таблиця 4 – Таблиця прийнятих ІДС рішень

Процес	№ альтернативи	Ресурс	Дохід
Процес 1	1	0	5
Процес 2	3	2	12
Процес 3	1	0	3
Процес 4	2	1	7
Процес 5	3	2	6
Процес 6	2	1	7

Висновки

У даній статті запропонована архітектура інтелектуальної системи інформаційного пошуку, яка використовує як критерій релевантності текстового документа приріст корисності стану агента, що отримав інформацію, яка міститься у цьому документі, відносно стану, коли зміст документа агенту ще не був відомий. З цією метою розглядається метод оцінки функції корисності природомовного тексту на основі його модельного представлення у формі повідомлення. Проведено аналіз стану досліджень та розробок у галузі інтелектуальних пошукових систем. Засобами RDF-графів формалізовано представлення операцій з побудови оптимального плану інтелектуального агента як моделі інформаційних потреб клієнта для оцінювання цінності нових знань.

На прикладі використання онтології матеріалознавства показано схему модельного представлення процесу побудови оптимального плану відновлення (протикорозійного захисту) нафто-газопроводу. Для цього використовується інформація про альтернативи підпроцесів основного процесу, їх ресурсозатрати та ефективність дії (наприклад термін експлуатації). Оскільки основний процес є керованим, то в результаті отримаємо задачу динамічного програмування, однак розроблений підхід є ефективним і для некерованих процесів – у такому випадку отримаємо марківську модель процесу. У такому випадку на базі онтології обчислюватимуться ймовірності пере-

бування об’єкта у різних станах. Під час машинного навчання онтології матеріалознавства за даними інформаційного пошуку в мережі Інтернет, яке полягає у пошуку нових альтернатив дій інтелектуальної системи, отриманий виграш у ефективності модельної реалізації стратегії відновлення-захисту трубопроводу забезпечує систему пошуку інформації чисельною оцінкою її корисності для клієнта.

Для реалізації онтології матеріалознавства обрано Protege-OWL. Модуль обчислення затрат ресурсів та прогнозований виграш написаний з використанням Java Protege-OWL API. Розроблена модель системи може бути впроваджена для автоматизованого інформаційного пошуку у тих проблемних областях, для яких побудована і навчена (натренована описаними методами і засобами машинного навчання) онтологія, яка виступає в ролі моделі інформаційних потреб користувача такої системи.

Література

1. Рассел Стюарт. Искусственный интеллект: современный подход / Стюарт Рассел, Питер Норвиг ; пер с. англ. – [2-е изд.]. – М. : Издательский дом «Вильямс», 2006. – 1408 с.
2. Ontology-Based Meta-Mining of Knowledge Discovery Workflows / Melanie Hilario, Phong Nguyen, Huyen Do [и др.] // Studies in Computational Intelligence. – 2011. – Volume 358/2011. – P. 273-315.
3. A Knowledge Engineering and Planning Framework based on OWL Ontologies / Eric Bouillet, Mark Feblowitz, Zhen Liu [и др.] // Knowledge Engineering Review. – 2004. – Vol. 18:3. – P. 209-220.
4. Beckett, D. 2004. Rdf/xml syntax specification. http://www.w3.org/TR/rdf-syntax-grammar
5. Baresi L. Tutorial introduction to graph transformation: A software engineering perspective / L. Baresi and R. Heckel // In 1st Int. Conference on Graph Transformation. – 2002.
6. Riabov A. Scalable planning for distributed stream processing systems / A. Riabov and Z. Liu // In ICAPS’06. – 2006.
7. Description logic programs: combining logic programs with description logic / B. Grosof, I. Horrocks, R. Volz and S. Decker // In WWW’2003, (May 20 – 24, Budapest, Hungary). – 2003.
8. Noy N. Defining n-ary relations on the Semantic Web [Електронний ресурс] / N. Noy and A. Rector. – Режим доступу : http://www.w3.org/TR/swbp-n-aryRelations/
9. Lytvyn V. Intelligent agent on the basis of adaptive ontologies construction / V. Lytvyn, D. Dosyn, M. Medykovskyj, N. Shakhovska // Signal Modelling Control, (27 – 29 June 2011, Lodz). – 2011.

Literatura

1. Rassel, Stjuart, Norvig, Piter Iskusstvennyj intellekt: sovremennyj podhod, 2-e izd. Per s. angl. M.: Izdatel'skij dom “Vil’jams”. 2006. 1408 s
2. .Melanie Hilario, Phong Nguyen, Huyen Do, Adam Woznica and Alexandros Kalousis / Ontology-Based Meta-Mining of Knowledge Discovery Workflows / Studies in Computational Intelligence, 2011, Volume 358/2011, 273-315.
3. Eric Bouillet, Mark Feblowitz, Zhen Liu, Anand Ranganathan, Anton Riabov /A Knowledge Engineering and Planning Framework based on OWL Ontologies
4. Beckett, D. 2004. Rdf/xml syntax specification. http://www.w3.org/TR/rdf-syntax-grammar
5. Baresi, L., and Heckel, R. 2002. Tutorial introduction to graph transformation: A software engineering perspective. In 1st Int. Conference on Graph Transformation
6. Riabov, A., and Liu, Z. 2006. Scalable planning for distributed stream processing systems. In ICAPS’06
7. Grosof, B.; Horrocks, I.; Volz, R.; and Decker, S. 2003. Description logic programs: combining logic programs with description logic. In WWW
8. Noy, N., and Rector, A. 2004. Defining n-ary relations on the Semantic Web. http://www.w3.org/TR/swbp-n-aryRelations/
9. Lytvyn V. Intelligent agent on the basis of adaptive ontologies construction / V.Lytvyn, D.Dosyn, M.Medykovskyj, N.Shakhovska // Signal Modelling Control. – Lodz. – 27–29 June 2011.

RESUME

D.G. Dosyn, V.M. Kovalevych

An Architecture of an Intelligent System of Information Search on the Internet

In this article the architecture of intelligent information search system, using as a criterion of a text document relevancy an agent state utility increasing after receiving by them an information contained in such document with respect to the state where the content of the document was not known. For this purpose, the method of estimation of the utility function natural language text based on its model representation in the form of message.

An analysis of research and development in intelligent search engines was performed. By means of RDF-graph representation formalized operations building intelligent agent optimal plan as a model the information needs of the client to assess the value of new knowledge.

On the example of materials science ontology show model representation of the process of building the optimal pipeline recovery (corrosion protection) plan. It uses information about alternative sub processes of the main process, costs of resources and effectiveness (e.g. lifetime prolongation). Since the basic process is controlled, as the result we obtain a dynamic programming problem, but developed an approach is effective also for uncontrolled processes - as we obtain Markov model of the process. In this case, probabilities of the object stay in different states will be computed using the ontology data.

During the process of materials ontology machine learning using the information that is received through the Internet, which is consist of seeking new alternatives for local tasks solving actions for intelligent agent, the performance shift of the pipeline recovery-protection strategy provides the information search system by numerical value of benefit for the customer.

Protege-OWL was selected to implement the materials science ontology. Module of computing of spent and obtained resources was written using Java Protege-OWL API. The model system can be implemented for automated information retrieval in those areas of concern, which built and trained using described methods and means of ontology machine learning such as this ontology serves as a model of a system user information needs.

Стаття надійшла до редакції 05.06.2012.